1972

Hidden-line elimination with interactive graphics display considerations.

Poggi, Steven Neil.

Monterey, California. Naval Postgraduate School

HIDDEN-LINE ELIMINATION WITH INTERACTIVE GRAPHICS DISPLAY CONSIDERATION

Steven Neil Poggi

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

Hidden-Line Elimination with Interactive Graphics Display Considerations

by

Steven Neil Poggi

Thesis Advisor: George A. Rahe

June 1972

Approved for public release; distribution unlimited.

Hidden-Line Elimination with Interactive Graphics Display Considerations

by

Steven Neil Poggi
Lieutenant, United States Navy
B.S., University of California, Los Angeles, 1966

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1972

ABSTRACT

Problem areas in the field of computer graphics, as applied to three-dimensional space, are introduced through a discussion of hidden-line elimination and perspective views. An adaptation of a simple, fast algorithm for hidden-line elimination is presented.

Graphics terminal display capability for a two-dimensional representation of a surface is made available. Interactive extensions to the basic program are developed to enhance potential applications for design and analysis.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENT

The author wishes to thank Mr. William A. Thomas of the Naval Postgraduate School Computer Laboratory for his systems programming assistance.

I. INTRODUCTION

Man-machine communication of three-dimensional information pertinent to shape descriptions has been, in recent years, a subject under investigation. New techniques to facilitate design and analysis activities have been adopted using graphical methods.

A. THE GENERAL PROBLEM

The implementation of translation and rotation capability to include hidden-line removal for three-dimensional objects in perspective views is indeed a formidable problem. The goal, a procedure generalized to include a large class of arbitrary surfaces, objects, and space-curve intersections, has long eluded investigators in this field.

The difficulties encountered in this area have been compounded by the processing time requirements of generalized procedures. Approaches taken in the solution of the general problem have primarily been to narrow or isolate one or more sub-problem areas for analysis.

B. THE SPECIFIC PROBLEM

In order to limit the extent of investigative efforts in relation to the general problem, a more specific goal was developed. Graphical display output of a surface or series of curves to include hidden-line elimination was the primary problem area to be investigated. A constraint on processing time requirements was added for real-time analysis capability. Exploitation of the highly dynamic characteristics of interactive computer graphics was also to be investigated. Operator modification of the displayed structure and selection of design and analysis options were seen as potential candidates for implementation.

With generalized objectives in mind, a potential field for application was sought in order to further specify the nature of the techniques to be developed. Signal processing provided a representative area for possible application of resulting methods. An example of the three-dimensional display structure required was contrived. A series of curves, each curve to represent a frequency-amplitude function at a point in time would be exhibited. In terms of rectangular coordinates, the ordinate might represent frequency, the abscissa depict signal amplitude, and time as the depth dimension. An example of a processed signal represented by such a display structure is shown in Figure 1; with separate curves corresponding to the signal at discrete points in time.

To facilitate analysis of the display, hidden-line elimination was deemed essential. Visual determination of relative amplitudes among succeeding curves would be difficult, at best, with wire-frame representation.

A real-time display environment was part of the problem specification, thereby adding a processing time constraint. Relating this condition to the signal processing display example, time delays between drawing of the curves should be minimized. Since each curve would represent signal sampling at a discrete point in time, visual interpolation between curves by a viewer would provide the concept of a surface, or continuum of sampling.

Rotation of the displayed structure was desired to present the user with optional aspects, or viewing angles.. For example, in Figure 1, the signal ridge would not be visible if the display were oriented (rotated) with receding curves to the left relative to the initial signal curve.

Figure 1. Example of Display Structure.

Perspective views of a family of curves were not initially considered for investigation. It was felt that qualitative comparisons would depend on visual reference among similar curves, i.e., not transformed to produce perspective depth cues for three-dimensional objects. Offset in viewing angle by simulated rotation was anticipated to provide the necessary time-step (depth) discrimination.

Some degree of interaction between an operator and a graphics display unit linked to a computer was demanded. The potential of a cathode ray tube (CRT) for display would not be fully used without regard to man-machine communication links. The express form of interaction was not specified. However, enhancement of quantitative analysis and various display options were areas for investigation.

No mention has yet been made of the data format requirements for the display. The intention was to provide a highly useful output medium. Any preprocessing of data to provide a suitable format for manipulation and graphical display was assumed.

C. PREVIOUS WORK

Following are representative summaries of research to date in the field of object graphics display. Hidden-line removal and perspective view considerations are emphasized.

Johnson [Ref. 8] presented a report including a good background for his then present activity, the manipulation of straight-line, wire-frame figures in three-dimensional space. The visual presentation consisted of three orthogonal views in addition to a perspective view of an object. Johnson states that the two-dimensional views without perspective would not convey the proper depth information. The perspective view alone also did not visually display all the correct information, particularly with regard to wire-frame representation. Johnson also chose to emphasize the importance of real-time, bilateral communication, as did Sutherland [Ref. 21]. The immediate future problems, as they appeared to Johnson, were allowance for arbitrary surfaces, surface intersection determination, and hidden-line elimination.

An article by Puckett [Ref. 20] considered rotation and translation of three-dimensional objects prior to perspective projection on a two-dimensional surface. Stereographic pair applications were also discussed. Again, wire-frame representation imposed limitations on the usefullness of the display.

More recent work, primarily concerned with perspective representations, has come to include consideration of hidden-line removal as an integral part.

Kubert, Szabo, and Giulieri [Ref. 9] introduced their investigations of functions of two variables by discussing contour plots and planar projections. Contour plots had an inherently quantitative nature. Planar projections were used where qualitative descriptions were essential. Families of curves used to define an object were subjected to prespective transformations. Criteria for the visibility of points defining an object were established.

Galimberti and Montanari [Ref. 5] had shifted the emphasis to hidden-line elimination in their investigations of perspective display of concave and convex plane-faced objects. Computing time proved to be a major constraint.

An earlier article by Appel [Ref. 1] included a discussion of previous research into hidden-line determination and elimination. Limitations prior to that time had been on the types and shapes of objects considered. With point visibility schemes applied to complex objects, calculation times had increased to unacceptable levels for many applications. Appel introduced his idea of "quantitative invisibility" to reconcile surface and point visibility approaches.

Freeman and Loutrel [Ref.4] were concerned with the two-dimensional version of the general hidden-line problem. This limitation was imposed in order to work with arbitrary, irregular polygonal figures.

Comba [Ref. 2] addressed the problem of detecting intersections of three-dimensional objects mentioned earlier by Johnson [Ref. 8].

Included was a discussion of the overall hidden-line problem as well as several interrelated sub-problems. Prior efforts by other investigators were mentioned.

A recent contribution by Matsushita [Ref. 10] used initial hidden-line computations for subsequent calculations involving rotation about a single axis. Reduction of processing time requirements was the result. As seen in earlier research, computation time constraints were proving to be a major motivating force for work undertaken in pursuit of problem solutions to real-time applications.

The reader is directed to a comprehensive review of perspective view and hidden-line elimination considerations by Desens [Ref. 3]. Included in the thesis was an extensive bibliography.

A pair of articles dealing with hardware aspects in relation to the problem areas at hand should be mentioned.

Hagan and Treiber [Ref. 6] provided an interesting approach to generalized linear transformation, spatial coordinate transformation, generation of visual displays for three-dimensional objects in perspective, generation of functions of two or more variables, and convolution and correlation operations. The authors described a number of hybrid analog-digital techniques using multiple analog and hybrid elements in parallel.

A hybrid, parallel array, composed of digital-to-analog converters (DAC's) and hybrid multiplying elements (HME's), were used for rotation, translation, and perspective representation of an object. This spatial coordinate transformation was performed on each point defining an object. The authors stated that using an analog vector generator which develops a line segment in 30 microseconds, it was possible to display objects

consisting of about 1000 line segments at a frame repetition rate of 30 frames per second.

It was necessary to perform nine multiplications, 12 additions, and two divisions for each operand fetch of the values representing a point. The parallel nature of array elements reduced the number of arithmetic operations per operand fetch to slightly more than seven. These developed techniques had proven useful for a wide range of real-time, signal processing applications.

Hagan, Nixon, and Schaefer [Ref. 7] presented background motivation for design of the Adage line of graphics terminals. Rotation, depth cueing, and windowing techniques were specifically discussed in terms of hardware implementation. Basic software operators were also mentioned.

II. EXPERIMENTAL PROCEDURE AND RESULTS

Experimentation commenced with programming to implement a graphical display, following reference research into previous work done on the problem to be investigated here.

A. GRAPHICAL DISPLAY OF THREE-DIMENSIONAL OBJECTS

Programming and test runs to allow display of three-dimensional objects were attempted first. The concepts discussed and techniques developed by Desens [Ref. 3] formed the basis of a program. One of the basic goals of the program was to transform the rectangular coordinates of an object into two-dimensional perspective for CRT display.

A single 4 x 4 matrix, [T], performs rotation, scaling, translation, and perspective transformations. Partitioning of [T] separates those four operations. The upper left is a 3 x 3 rotation matrix. The first three elements of the bottom row hold the translation terms. The first three elements of the right column contain inverses of the three viewing distances necessary for perspective transformation. The bottom right element is a single scale factor.

The matrix operates on a homogeneous coordinate vector. Original, orthogonal coordinates (x,y,z) become the homogeneous coordinates (x_h,y_h,z_h,s_h) where $x_h = s_h x$, $y_h = s_h y$, and $z_h = s_h z$. The added, fourth coordinate, s_h, is considered a variable scale factor.

Transformation of a point is made by post-multiplying a homogeneous coordinate vector by the complete transformation matrix, $[x_h,y_h,z_h,s_h]$ $[T] = [x_h',y_h',z_h',s_h']$. To obtain the final display coordinates for the

X-Y plane, division of the transformed homogeneous coordinates by the s_h' coordinate is made, that is $X = x_h'/s_h'$ and $Y = y_h'/s_h'$.

The program included a two-stage procedure for removal of hidden-lines. First, all lines hidden by an object's own volume were removed. Next, each segment not yet eliminated was examined over it's length and that portion remaining in view was displayed.

Two test objects were used to supply data points for experimental computations. An L-shaped block defined by 12 points was first displayed. The second test object, an aircraft carrier, consisted of 62 points. Each data point was specified in rectangular coordinates.

The results from tests were successful in terms of the resulting graphics display output. However, the preliminary results were not satisfactory from two standpoints. While the perspective view transformations provided the proper depth cueing, the hidden-line elimination segments of the program used were not functioning currectly. Additionally, processing time requirements were excessive for a simple and a moderately complex object, i.e., the two test objects.

Confirmation of Desens' conclusion [Ref. 3], regarding the relationship between processing time requirements and picture complexity, was made on the basis of the test runs. Picture complexity was defined by the number of plane-faced surfaces included in an object and thus the number of lines defining those surfaces. The conclusion validated by the test runs was that time required for connection of points increases rapidly as picture complexity increases, approximately in proportion to the square of the number of lines.

B. HIDDEN-LINE ELIMINATION

A different approach to the hidden-line removal problem was taken next. A simple, fast algorithm with emphasis on hidden-line elimination was sought. Attention was directed to the hidden-line plotting program developed by Williamson [Ref. 22].

1. The Basic Program

Subroutine HIDE, Appendix C, yielded a two-dimensional representation of a surface or object. Curves or segments of curves were plotted where not hidden by segments previously plotted. Any curve segment lying above a visual maximum function was determined to be visible.

Options included in the program were translation of data points to simulate stepping in the depth dimension, plotting of a border, axes, and a title, and plotting of the visible portion of the underside of a surface. In the latter case, segments of curves were presumed to be hidden where they lay above those segments previously plotted. Provisions for perspective view or rotation transformations were not included in the program. The appropriate transformations could be applied prior to a call to subroutine HIDE.

The first call to HIDE was for initialization and output of the curve farthest in the foreground. Subsequent calls to subroutine HIDE performed hidden-line determinations and output of visible curve segments farther in the background.

The main program computes the x and y coordinates of the curves (function) to be displayed. Additionally, the calling program sets initial values for control of display options.

2. Extensions

The basic program was to be extended to encompass graphical display output and as many interactive options as possible within the limits imposed by time constraints.

a. Graphical Display

Adaptation of the program to graphics terminal display was the first step. Equipment to be used consisted of an Adage Graphics Terminal (AGT-10) and an SDS-9300 computer, both located in the Naval Postgraduate School Computer Laboratory.

The SDS-9300 is a medium sized, general purpose, digital computer system. The sophisticated central processing unit can handle a large number of operations. Main memory consists of 32K words of core. A magnetic drum is used for secondary storage. The system is controlled at the operator's console and at a teletypewriter. Input is via card reader, teletypewriter, paper tape reader, or magnetic tape. Output is sent to a line printer, paper tape punch, teletypewriter, or magnetic tape. Communication links are available to two Adage AGT-10's.

The AGT-10's are small, general purpose, digital computers with each having 4K words of main memory plus a magnetic disk unit for secondary storage. A graphics display console is linked to each AGT-10 for text and graphics presentation. Inputs to an AGT-10 are a teletypewriter, paper tape reader, lightpen, joystick, function switches, and variable control dials. Output appears on a CRT, paper tape punch, or teletypewriter.

Display capability was implemented through the use of systems graphics routines DGINIT [Ref. 16], IHEAD [Ref. 12], IPACK [Ref. 13], and GRAPHO [Ref. 18]. The purpose of FORTRAN callable subroutine DGINIT

is to initialize the AGT-10 graphics display subsystem. DGINIT must be called before any graphics communication may occur between the SDS-9300 computer and an AGT-10. Function IHEAD constructs the one-word header of a graphic data block to be displayed by a subsequent call to GRAPHO. One of the two parameters passed to IHEAD determines whether all the vectors drawn are to be solid lines or dashed lines. The other parameter specifies display intensity. FORTRAN callable function IPACK will pack the x and y values and a move-draw indicator for a coordinate point into one word of a graphic data block. Subroutine GRAPHO transmits a graphic data block from the SDS-9300 computer to an AGT-10 for display.

Scaling of data points was necessary for reconciliation of differences in the real number values used by the graphics terminal and the actual size or scale of the displayed object.

In order to maintain a reference point from which to work, data points were maintained in their original, unscaled format. A parallel data structure, array FIGURE, was created to provide the vehicle for storage of the scaled, packed data words.

The communication link between the SDS-9300 computer and an AGT-10 graphics terminal was established through AGT-10 systems program GATED. Before a user's program is executed on the SDS-9300 computer, GATED must be loaded and executed on an AGT-10. The purpose of GATED is to allow display (and optional editing) of graphics and/or text data blocks received from the SDS-9300. GATED is further described in more detail in Ref. 17.

The initial scheme for determining which data points were to be displayed consisted of witholding display of visible portions of a curve as they were computed until the visual maximum function had been

completely updated for the current curve, i.e., current call to subroutine HIDE. At that point the complete visual maximum function was transmitted to an AGT-10 for display.

Display in this manner, upon succeeding calls to subroutine HIDE, was unacceptable. Where hidden lines occured, portions of the visual maximum function were displayed more than once. This situation caused an accompanying increase in the intensity of certain curve segments. The gain in intensity was directly proportional to the number of times the same curve segment was displayed as a portion of the visual maximum function.

Changes were incorporated in the program to allow graphical display of only those portions of a curve lying above the visual maximum function. In this manner each visible curve segment was displayed only once, thus solving the problem of non-uniform intensities. However, a problem of storage requirements developed.

The interface systems program, GATED [Ref. 17], limited the number of graphics blocks allocated for display to 51. That was found too restrictive for the number of visible curve segments encountered in most aspects of the test function display. Anticipated requirements for future applications also exceeded the existing constraints imposed by GATED.

Consequent modification of GATED included allowance for more graphics blocks. The increase in storage required by the greater allowance of graphics blocks was more than offset by a coincident decrease in the number of text blocks.

An alternate solution to the problem of an insufficient number of graphics blocks for display of the entire test function involved

programming changes within subroutine HIDE. Instead of displaying each visible curve segment after being computed, display was witheld until all visible portions of a single curve were determined. The data points, representing the visible curve segments, were packed into consecutive locations in array FIGURE. Discontinuities between endpoints of segments were accounted for with a "move" indicator in function IPACK [Ref. 13]. A "move" rather than a "draw" was indicated for the vector from the last point of a visible segment to the first point in the next visible segment. Once all the segments were computed and packed, array FIGURE was displayed as a single graphics block. There then existed a one-to-one correspondence between the number of curves and the number of graphics blocks required. Visual results showed all the visible portions of a curve at the same instant. The preceding method had displayed each segment separately.

Consideration was given to display of the entire series of curves, representing the test function, at one time. This method would have involved, prior to display, storage of all visible points for the entire structure. The scheme used required temporary storage for display of each curve separately. While storage of this magnitude was available in the SDS-9300 computer, utilization was considered extravagant. Local variable storage requirements, in the case of the test function, would have increased to a total of approximately 12,000 words versus the 3,000 words required to include storage of one curve at a time. The increased storage would have been manifested in a highly expanded array FIGURE.

b. Test Function

The previously mentioned constraint imposed by subroutine GATED on the number of graphic data blocks also affected the selection of a suitable test function. The curves were consequently limited to 100 points each. This limitation enabled display of the test function with no discernible degradation in the smoothness of the curves. The number of curves chosen to represent a surface was 30. When this number of curves was combined with the appropriate translation values to simulate stepping in the depth dimension, a reasonable representation of a curved surface resulted, as in Figure 2. The number of data points, required by both the top and underside of the test function, caused a core overflow condition in the AGT-10. Additionally, a second call to HIDE was deemed to be unnecessary for the possible display features desired by the problem statement. For a series of two-dimensional curves offset for simulated depth stepping, i.e., the signal processing example (Figure 1), it was expected that coordinate points would be specified for positive half-space only.

Deletion of the second call to subroutine HIDE resulted in a significant decrease in processing time, since for even the hidden portions of the underside of the test function the majority of calculations in subroutine HIDE were made.

c. Interactive Extensions

Man-Machine interaction is a necessary and sufficient condition to justify the use of a graphics terminal. On-line modification of a display and program control flow options exercised by an operator sitting at a graphics terminal are highly desirable in many applications.

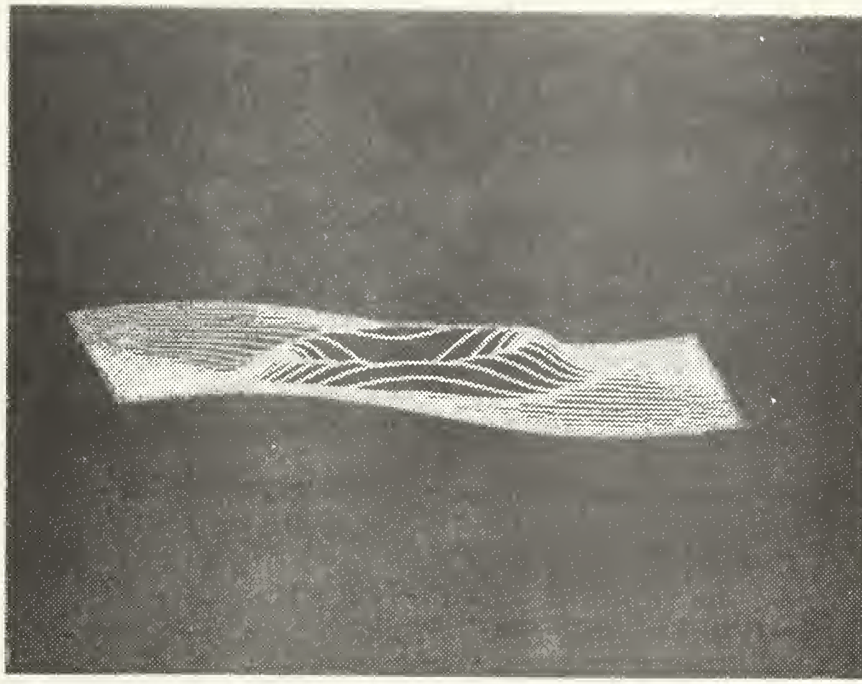


Figure 2. Test Function

(1) Variable Control Dials. The control dials were chosen as the method of input since they provided a range of values, not just simple switches. Also, they would maintain their settings until changed by the operator.

At the time experimentation was being conducted, no method existed for making available to the SDS-9300 computer, and thus to the FORTRAN program, the values of the variable control dials. While not a primary consideration of investigative work, control dial implementation resulted in a significant part of resource expenditures, time and effort.

The development of a utility program to use the control dial settings was accomplished. Appendix D contains a discussion of

implementation details and programming modifications. Appendix B includes instructions for on-line display modification and option selection using the variable control dials.

(2) Translation. A simple static translation operation was applied to the data points after scaling required for the graphics terminal display and prior to being sent across the communications interface to an AGT-10. This operation allowed an operator freedom in placement of a displayed structure on a CRT. Operator control of placement was input via variable control dial settings.

(3) Rotation. Rotation of the display structure about two axes was simulated. An apparent rotation of the surface represented about the y axis resulted from translating all points of succeeding curves in a negative x direction. In a similar manner, translation of all points of succeeding curves in a positive y direction resulted in simulated rotation of the surface about the x axis into positive half-space. Referring to Figure 3, the receding curves have been rotated (translated) to the left and up from the orientation in Figure 2.

Translation operations to simulate rotation were applied to arrays XH and H in conjunction with other modifications prior to graphical display. XH and H are working arrays initialized to the current visual maximum function in subroutine HIDE.

The apparent rotation effects produced by translations were acceptable for small to medium sized angles, i.e., up to approximately 40 degrees about the two axes. For larger angles, distortion of the displayed surface was seen. Part of the distortion could be explained by the fact that the curved surface was represented by a series of curves displayed close together and as the apparent rotation angle was increased,

the curves became more disjoint thus destroying the illusion of a continuous surface.

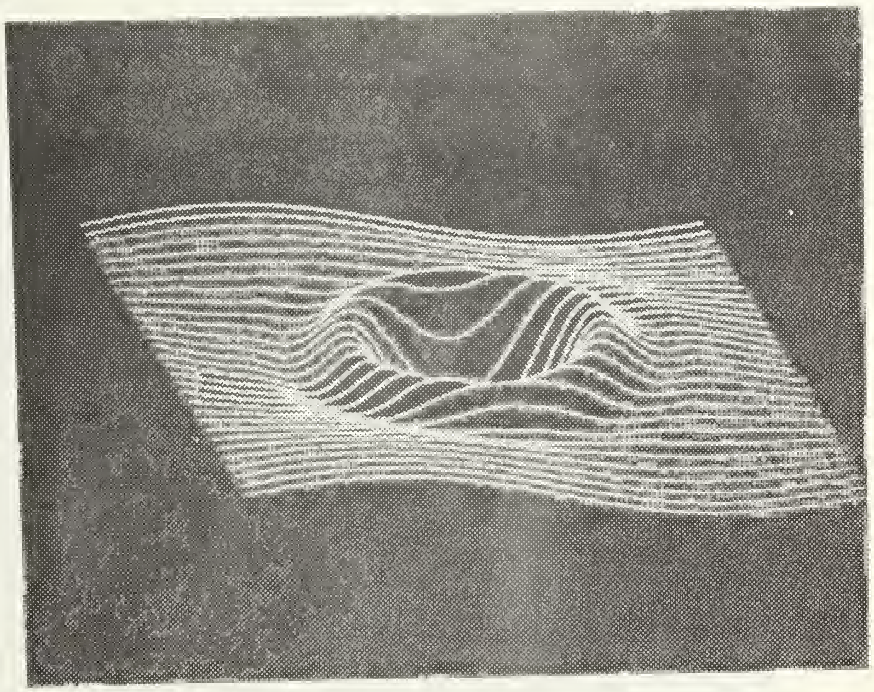


Figure 3. Rotation of Test Function.

Applications specifying a two-dimensional series of curves would not suffer from the display limitations imposed by this particular scheme as would three-dimensional object representation. For a series of curves, variable offset in the apparent viewing angle would be provided for possible qualitative trend analysis. The operator again had control over the variance of rotation (translation) by adjusting control dial settings.

(4) Window Operation. To enhance the capabilities of the developing program and to enable an operator to improve any qualitative

analysis of displays, a windowing operation was added. The window was envisioned as a planar slice through the displayed surface or family of curves. All points of the plane would have the same y value in three-space, i.e., parallel to a plane defined by the x and z axes (Figure 4).

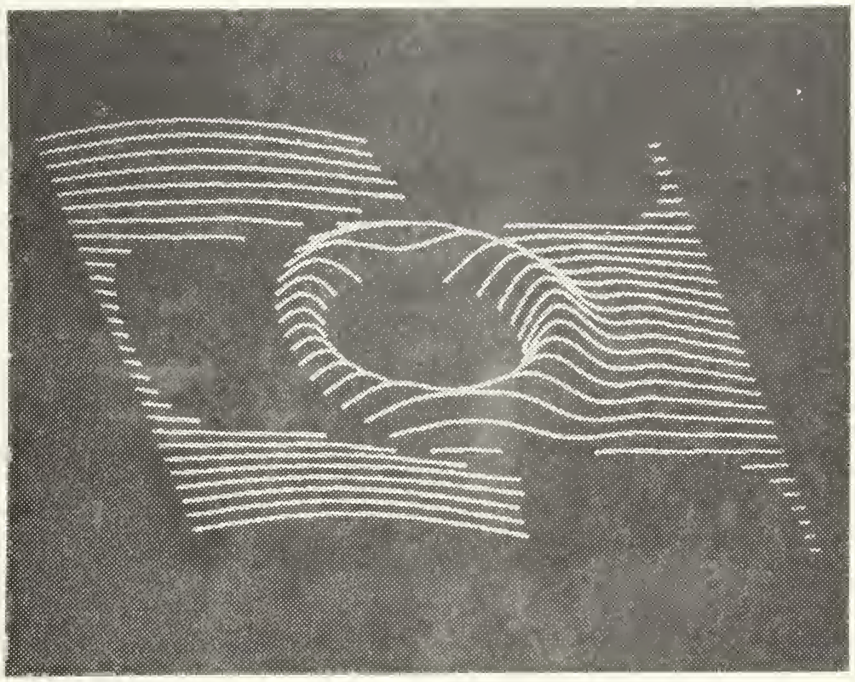


Figure 4. Windowing Operation.

In terms of programming, synthesis of the window was accomplished by merging a straight line, representing the window in relation to one curve from a series of curves, with the previous visual maximum function. A new visual maximum function was thus formed and was used for computing visible curve segments. The window acted as a lower threshold level, below which no curve segments were displayed. The

window "line" is translated (updated) for successive curves to provide the illusion of a planar slice in the same orientation as the display structure.

Modification was made to the program to include display of the first curve after hidden-line calculations were made. The previous concept had been that the first curve defined the initial visual maximum function for use with succeeding curves. With the introduction of a window operation, that concept was not valid. In other words, a portion of the first curve may also be hidden by the window (Figure 4). Operator control of the window was implemented with variable control dials.

(5) Cursor. Provision for some method of quantitative analysis was desired. Addition of a cursor was the next goal. As envisioned, the operator would be able to select the cursor for any of the series of curves. The cursor should be adjustable along the ordinate and of varying, controlable length to specify points on a curve for measurement. Also, placement of the cursor at points other than on a curve might be useful for determining relative points on different curves. The operator should have the option of suppressing, at any time, a cursor mode of operation, since the continuation of successive curves being displayed would be held pending the end of cursor analysis. The accuracy of data point location by cursor placement was not expected to be of a high degree due primarily to discrimination limitations. Operator matching of cursor and curve points was a crude, viusal process.

The cursor, as implemented for display, consisted of a single vector defined by two endpoints. The first, or reference, endpoint had a y value corresponding to a value of zero for the particular curve being analyzed. The second, or variable, endpoint had an operator

controlled y value. The x values for both endpoints were equal and both were determined by the operator.

In Figure 5, the reference endpoint is at the top and the variable endpoint has been adjusted to lie on the last curve displayed. The variable endpoint position represents a negative y value, lying below the reference endpoint.

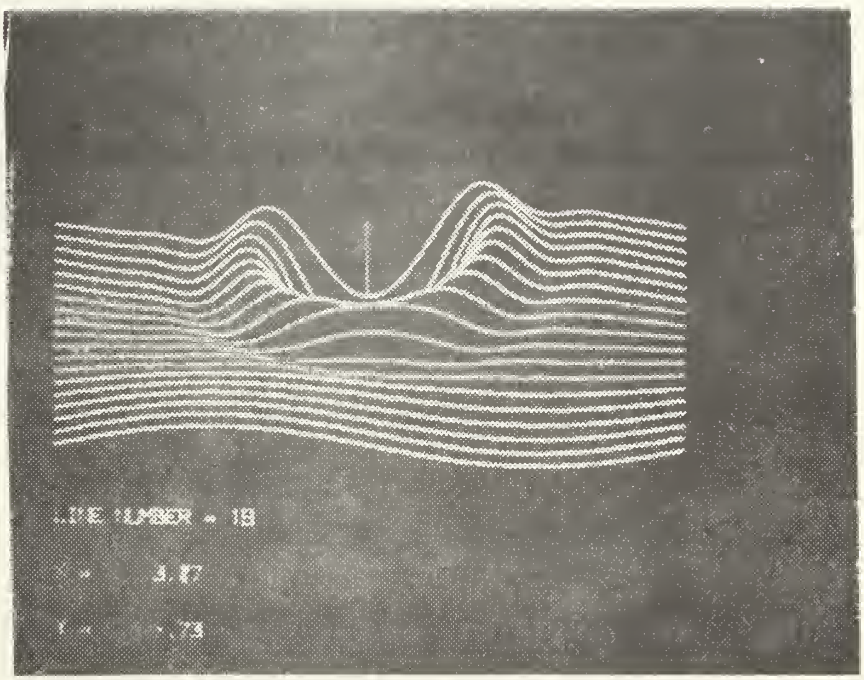


Figure 5. Cursor with Text.

The cursor could be invoked by an operator using a particular setting for one variable control dial. Once the cursor was displayed, a different dial controlled the ordinate values, and therefore the position on the CRT of the cursor. Another dial determined the abscissa value of the variable cursor endpoint, and therefore the

length of the displayed cursor. Still another dial value was tested in the program to determine whether to continue in the cursor mode for the current curve or to resume display of succeeding curves.

Display of the next curve caused the cursor to be blanked from the CRT since the next curve used the same graphics data block number as that used by the cursor. Output of the x and y values of the variable cursor endpoint were written on a line printer. The option to print the values was provided to the operator through another variable control dial. A call to systems subroutine DELAY was necessary to prevent the same values from being printed several times while an operator was modifying the dial settings.

While the basic implementation of the cursor seemed acceptable in terms of analysis capabilities, the remote output of variable cursor endpoint values was not adequate. Additionally, control dial manipulations were awkward.

Modifications to subroutine CURSOR were undertaken to eliminate the two deficiencies mentioned. Textual display of variable cursor endpoint values, appropriately scaled for the curve, would provide instantaneous feedback to the operator. The interactive nature of the analysis capability of the program would thus be increased.

Addition of a line number, displayed to aid the operator by providing a reference, was made. Display of the x and y variable cursor endpoint values were included in the same programming loop used to sense the control dial settings. By being in the same loop, only the value portions of the text were changeable once subroutine CURSOR was entered. Figure 5 is an example of cursor operation with accompanying text.

System subroutines used for text display were DTINIT and TEXTO [Refs. 11 and 15]. DTINIT initializes the AGT-10 text display subsystem and must be called before any text communications may be initiated between the SDS-9300 computer and an AGT-10. TEXTO allows output of a full (or partial) line of text on an AGT-10. Prior to a call to TEXTO, text characters are packed into a binary coded decimal, text array with four characters per word. The text array is then passed as one of several parameters to subroutine TEXTO. Formats for encoding of text characters are the standard FORTRAN I/O formats.

Return from the cursor mode of operation was now controlled by the adjustment of only one dial. Operator intervention, in terms of control dial manipulations, was significantly reduced.

III. CONCLUSIONS

Following is a discussion of summarized results and program extensions with potential applications in mind.

A. SUMMARY OF RESULTS

Graphical display implementation of a hidden-line removal algorithm has been accomplished. The program produces a two-dimensional representation (CRT display) of a three-dimensional surface or series of curves by displaying segments of a succession of curves where the segments are not hidden by any of the curves previously displayed.

Capabilities of the basic program were extended to enhance the design and analysis potential for applications use. Interactive attributes of the existing program include translation of the display structure for positioning, translation to simulate a limited rotational capability for presentation of different viewing angles, a windowing operation for designating threshold values of displayed curve segments, and a cursor mode of operation for analysis.

The operator is able to affect his desired changes in the display by input of variable control dial settings. An extended version of GATED, the systems communication interface routine, was implemented as POG3 to allow the operator this method of input.

B. DEVELOPMENT AND APPLICATIONS

Any extensions to the program, manifested as user options, will primarily be a function of specific applications. Reinclusion of a second call to subroutine HIDE will allow display of the unhidden part

of the underside of a curved surface. Changes to subroutine HIDE, as listed in Appendix C, will be necessary for consideration of the sign of the abscissa values displayed. The reader is cautioned to consider data point storage limitations and processing time increases caused by a second call to HIDE.

The option of displaying a border, axes, or title may also be desired. Scaled axes would provide a reference framework for estimating curve point values and orientation of the display as viewed.

Additional rotation capability is desirable. Rotation into quadrants, other than those presently allowed, will extend the present limitations imposed on viewing angles. The problem of figure distortion for large angles will, however, remain.

Calculations to result in perspective view display would significantly increase the class of possible applications. An example of the types of calculations necessary can be found in the transformation schemes used by Desens [Ref. 3] and Johnson [Ref. 8].

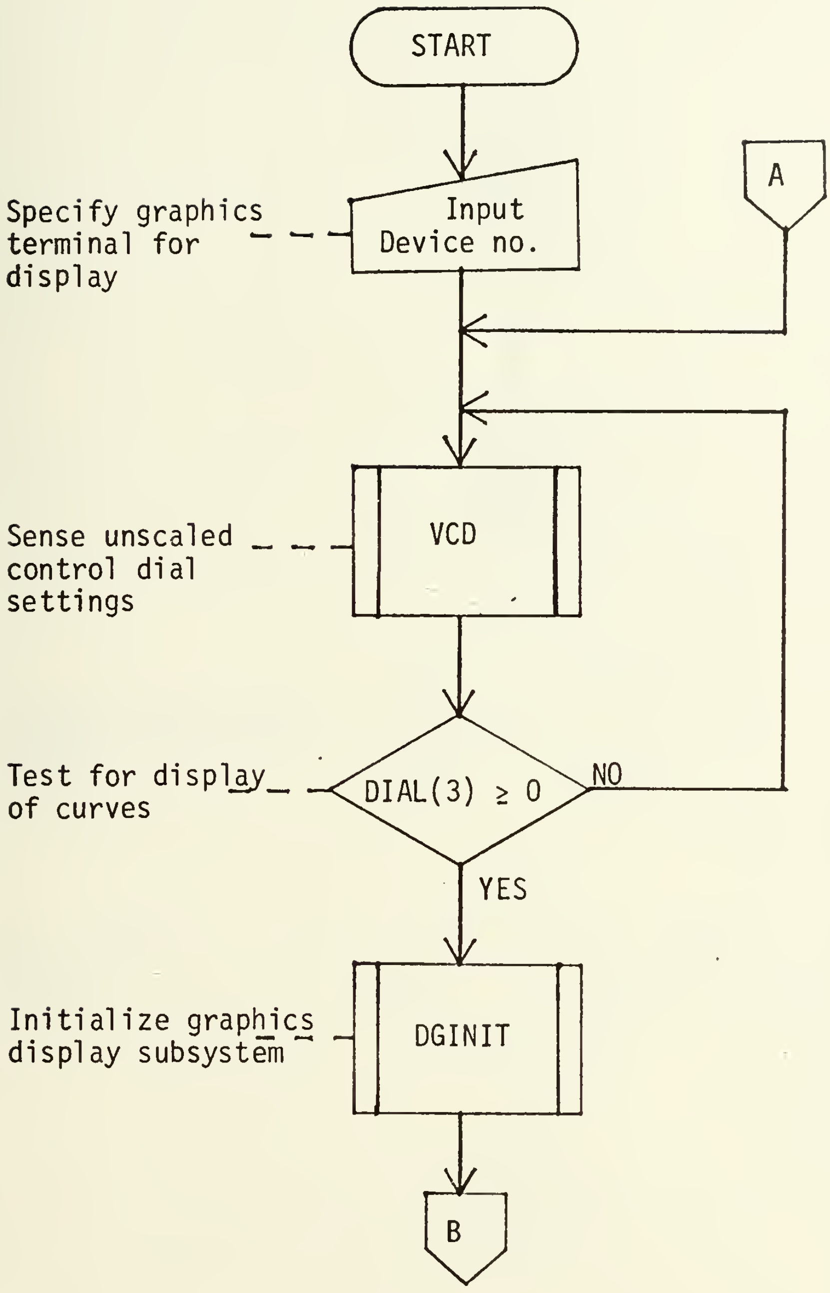
No consideration was previously given to text or graphics editing. These features are available with certain display devices. Editing of displayed structures may well be necessary for certain design applications.

As an alternative to design using editing features, it might be advantageous to allow the program user the ability to change parameters of a function, thereby modifying or creating his own figures. This capability could be implemented with additional control dials or similar input devices. Associated with parameter variation might be a "menu" selection option. The user would specify a basic function or object, for example with a lightpen, for later design modification through parameter variation.

The potential applications of hidden-line removal in the areas of design and analysis are many as amply documented in several of the references contained in the bibliography section. A simple, fast algorithm, as adapted here, is suited to a wide range of problems that call for display of real-time solutions. Interactive, man-machine communication is another essential ingredient of real-time applications.

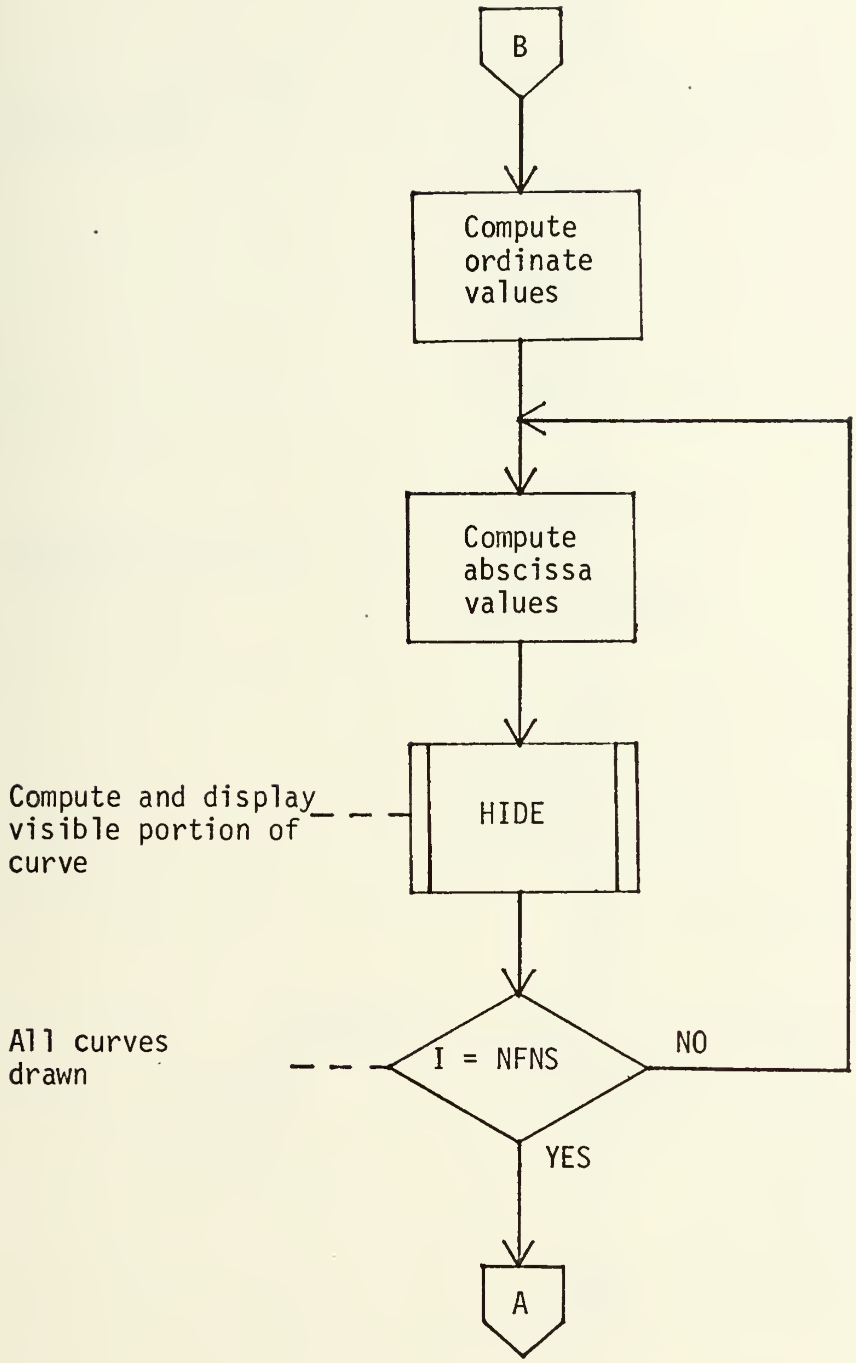
APPENDIX A

FLOWCHARTS

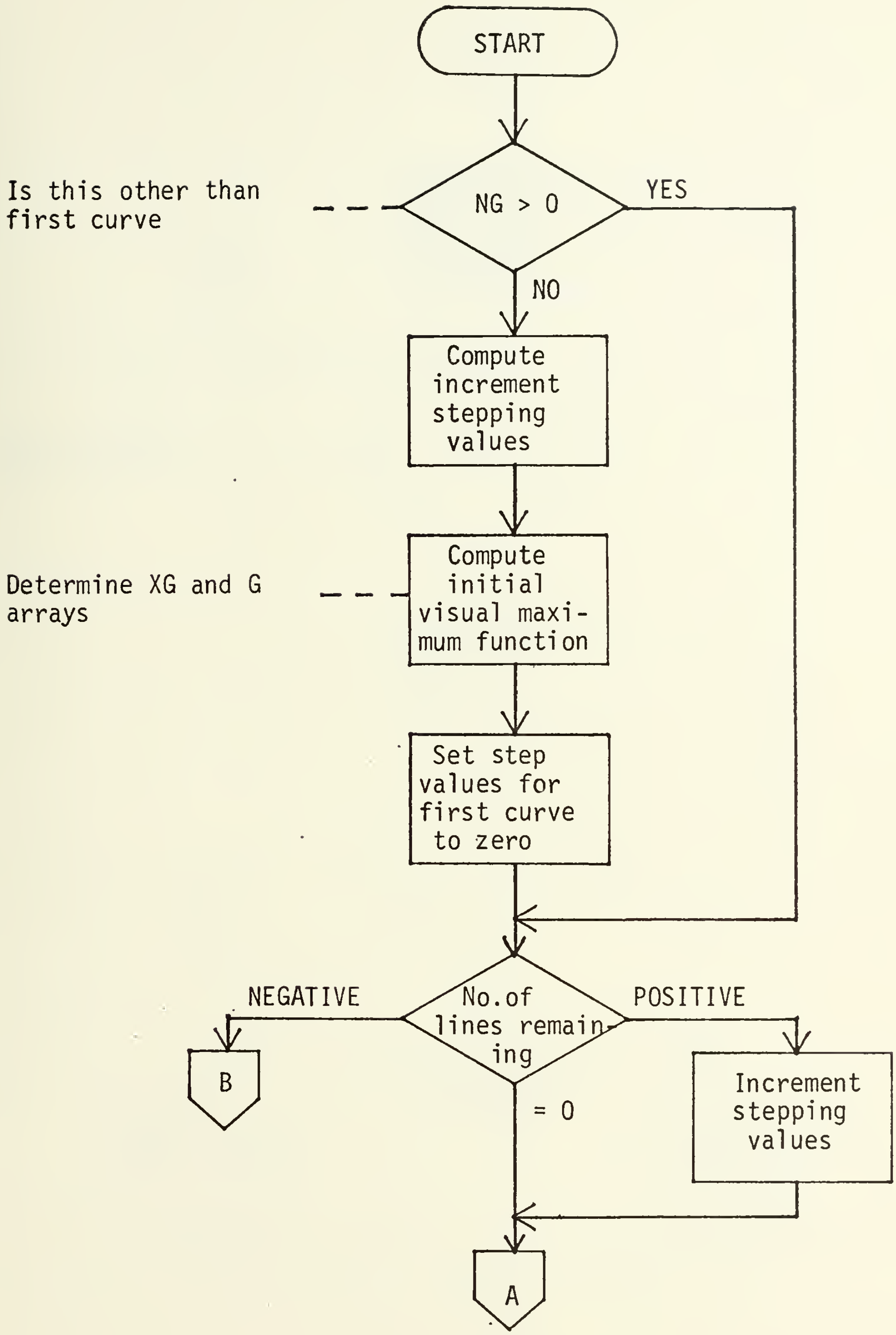


FLOWCHART 1

MAIN PROGRAM

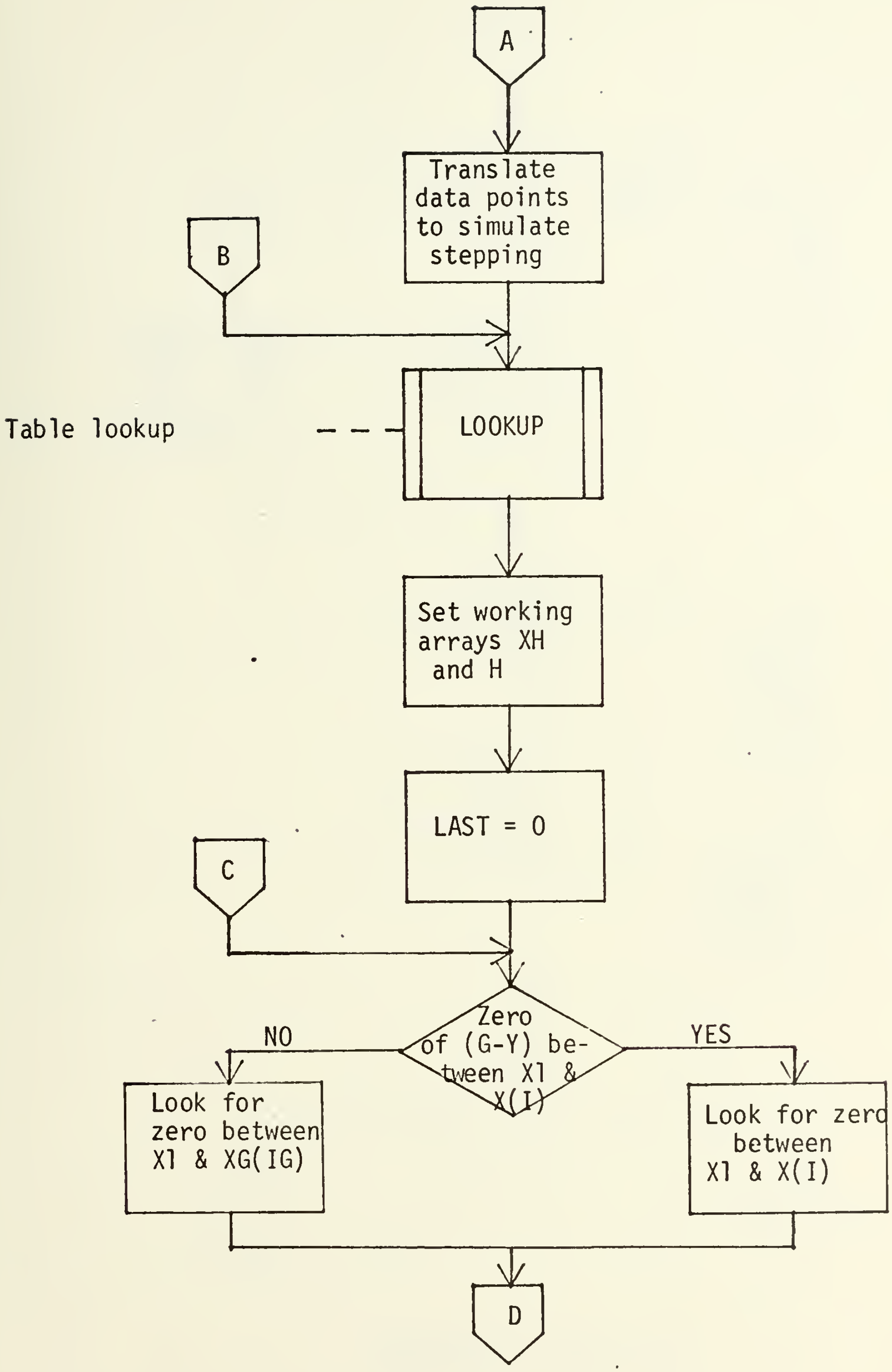


FLOWCHART 1 (Cont'd)

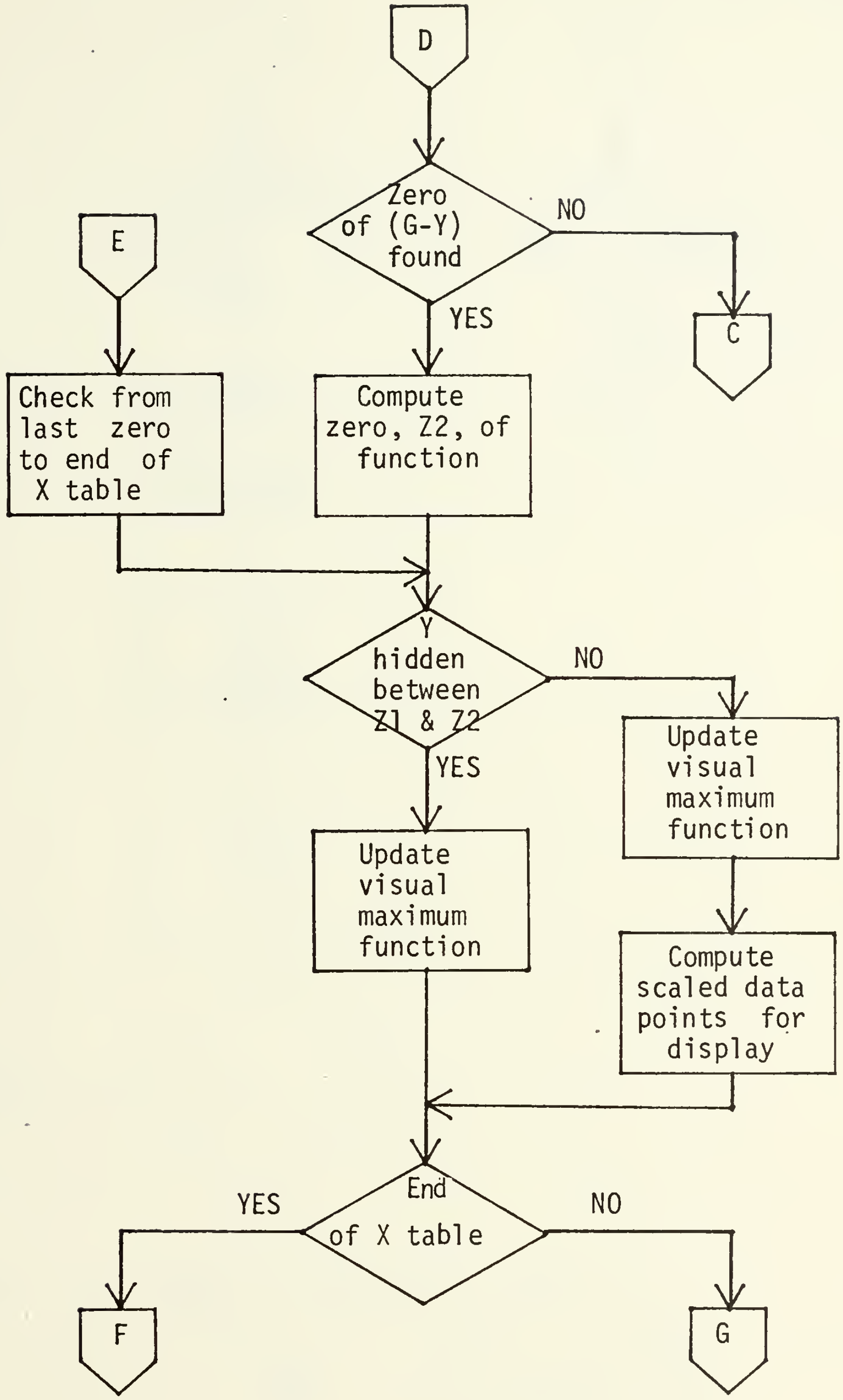


FLOW CHART 2

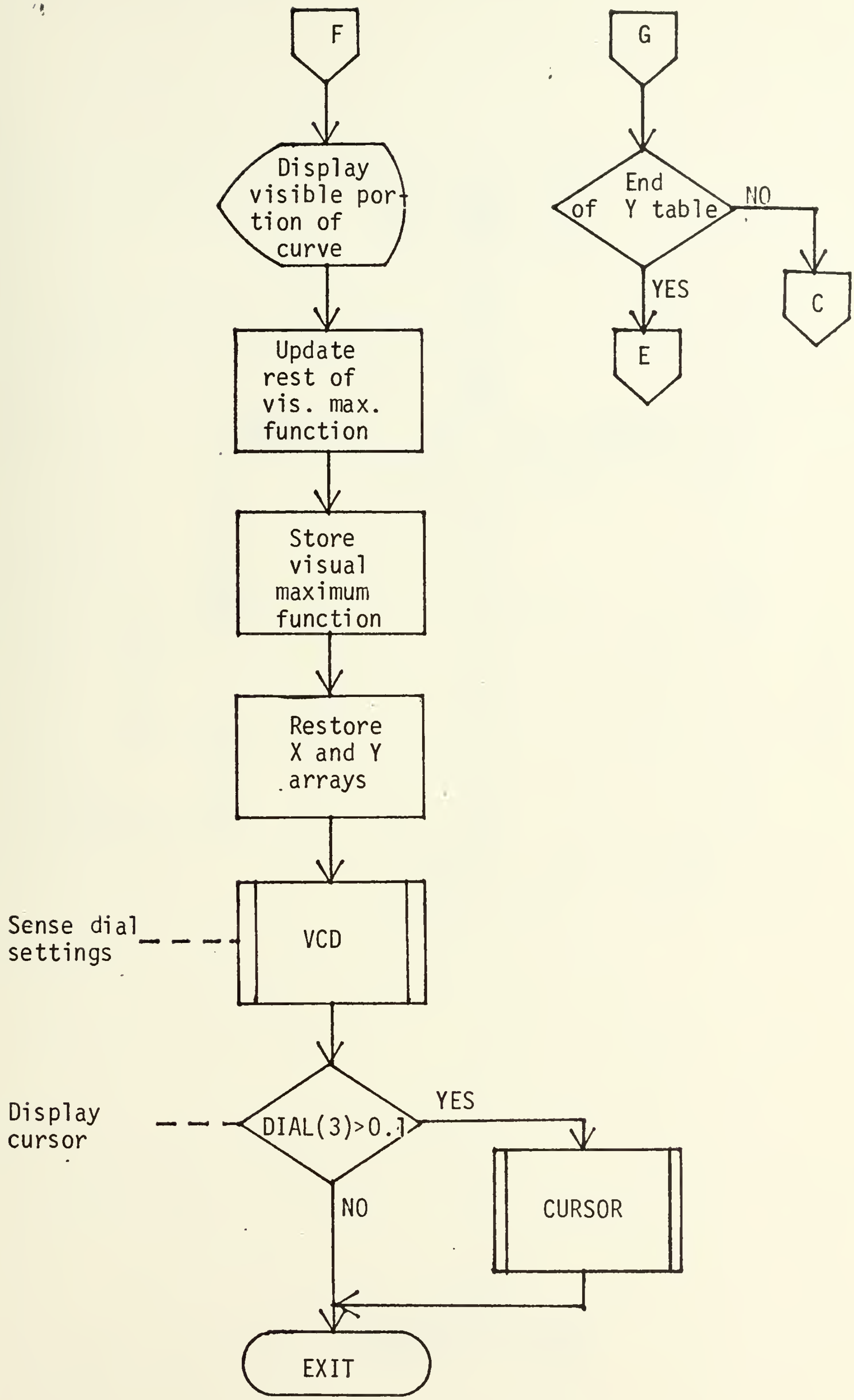
SUBROUTINE HIDE



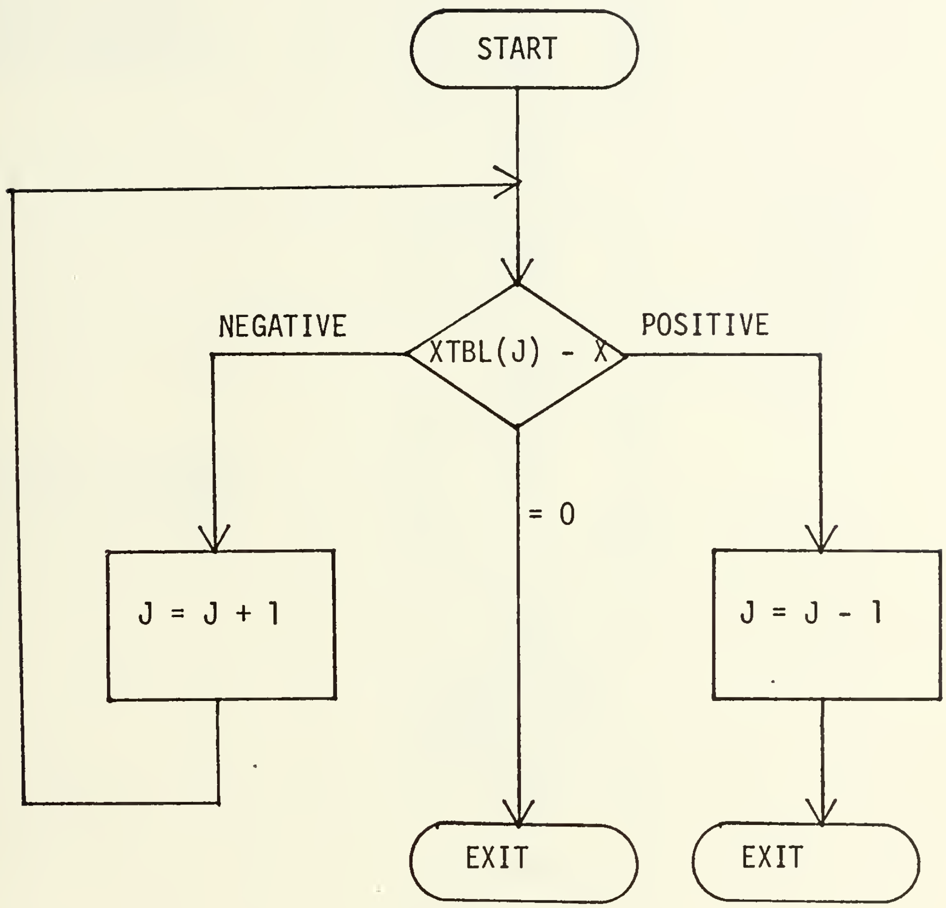
FLOWCHART 2 (Cont'd)



FLOWCHART 2 (Cont'd)

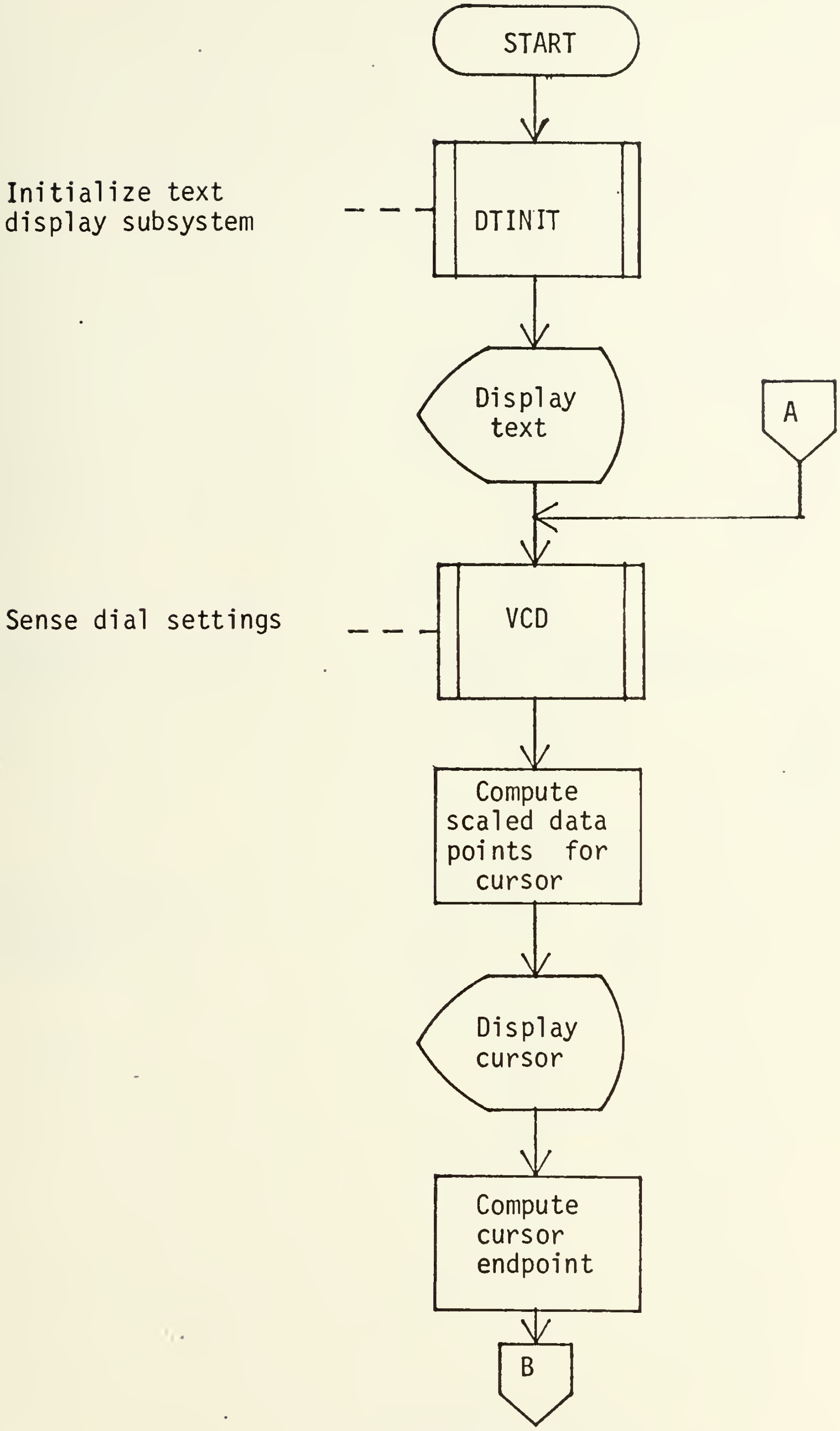


FLOWCHART 2 (Cont'd)



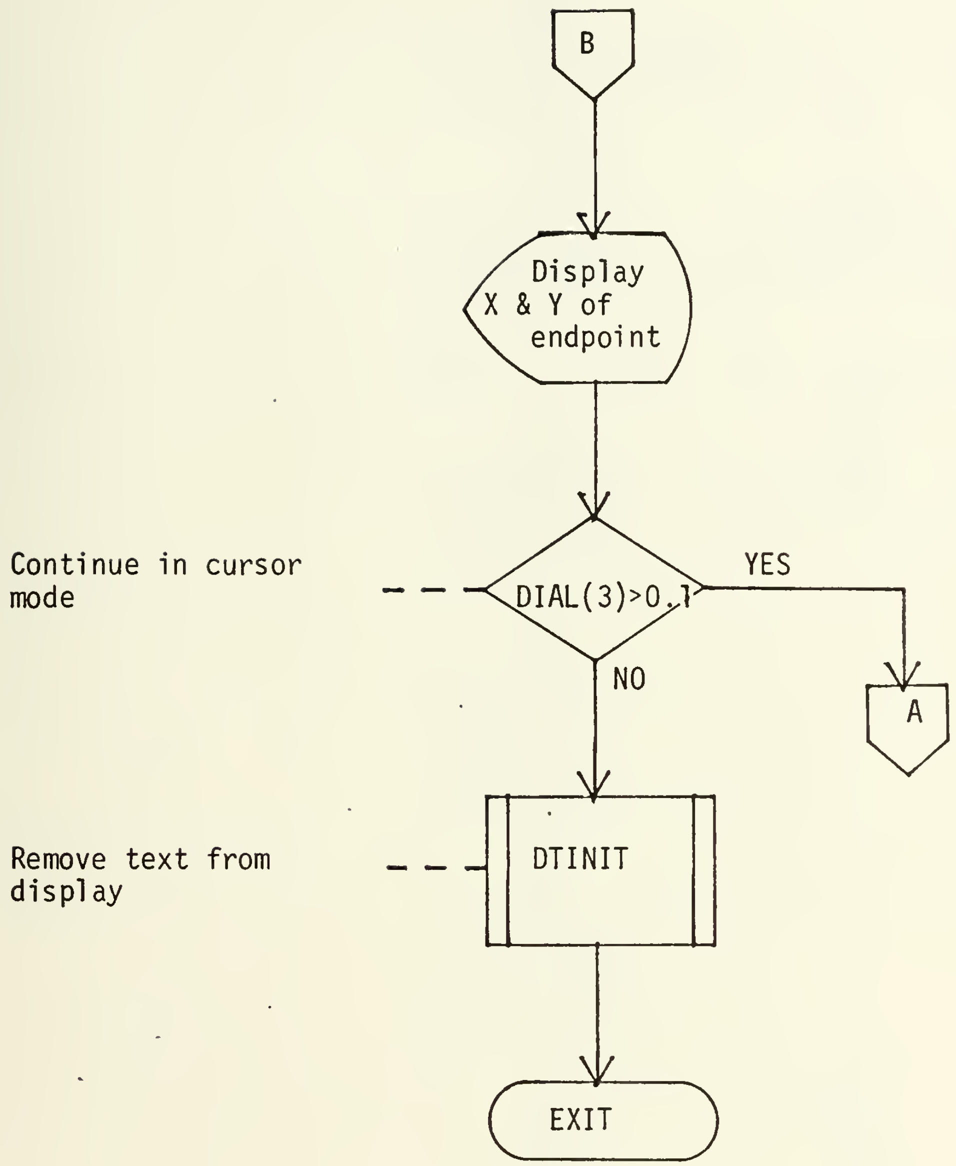
FLOWCHART 3

SUBROUTINE LOOKUP

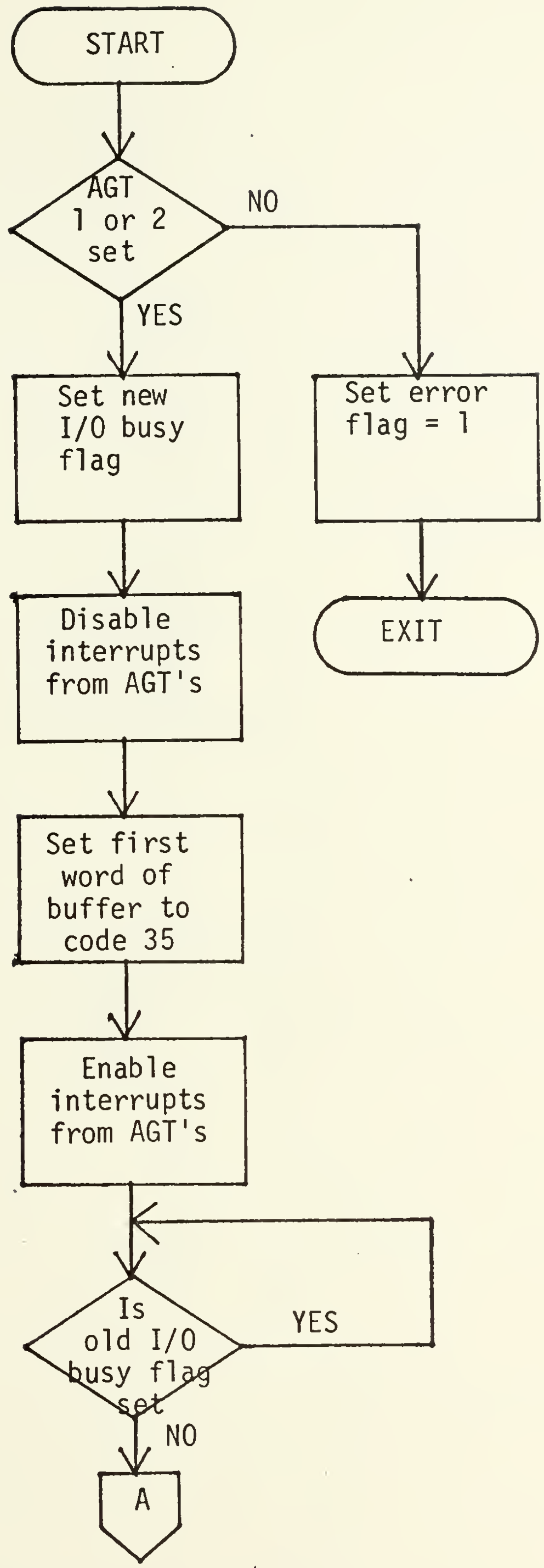


FLOWCHART 4

SUBROUTINE CURSOR

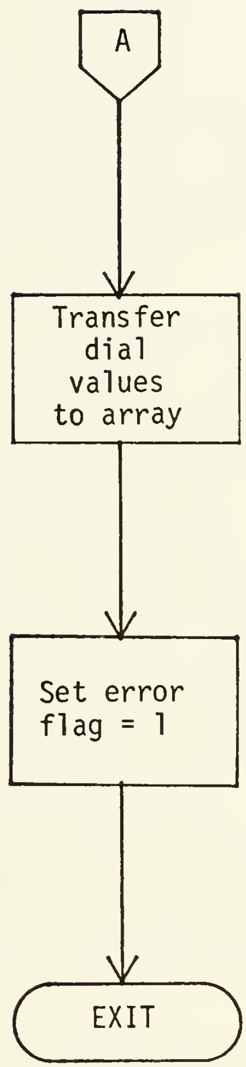


FLOWCHART 4 (Cont'd)



FLOWCHART 5

METASYMBOL SUBROUTINE VCD



FLOWCHART 5 (Cont'd)

APPENDIX B

USING THE PROGRAM

A. EXPLANATION OF VARIABLES

1. Main Program

X	The ordinate array for one curve.
Y	The abscissa array for one curve.
Y1	A working array used for intermediate values in computing Y values.
XG	The ordinate array of the visual maximum function.
G	The abscissa array of the visual maximum function
XH	A working, ordinate array used in conjunction with XG.
H	A working, abscissa array used with G.
IGDIR	The first word address of the graphics directory used as a parameter in calling DGINIT.
DIAL	The array used to hold unscaled values of the variable control dials.
MAXDIM	The dimension of arrays XG, G, XH, and H. MAXDIM is used for overflow condition testing of those arrays.
XMIN	The lower bound for X values.
DELTAX	The ordinate data increment.
YMIN	The lower bound for Y values.
DELTAY	The abscissa data increment.
XLNTH	The horizontal scaling factor used in computing translation.

YLNTH	The vertical scaling factor similarly used for translation.
XX	A constant used in computing Y values of the test function.
IDEV	Specifies which graphics terminal is to be used for display, either 1 or 2.
STEP	Has two meanings in main program. First, STEP is used as the ordinate value increment. Secondly, it is used as an incremental value in determining translation of succeeding curves.
TX	The scaled ordinate translation value for positioning of display
TY	The scaled abscissa value for positioning translation.
YWINDO	The scaled value of the window operator.
NG	The number of points in the current visual maximum function after each return from subroutine HIDE.
N	The number of points defining a curve to be connected by vectors for display.
NFNS	The number of curves defining a surface to be displayed.
NUMFIG	An index value used to keep track of sequential graphic data block numbers for calls to GRAPHO.

2. Subroutine HIDE

HDIAL	The array for local values of the variable control dials.

EPS1 The relative abscissa increment used to simulate discontinuities in the visual maximum function. EPS1 should be one or two orders of magnitude larger than the smallest recognizable relative difference in single-precision, floating-point arithmetic.

F(XX,XI,YI,XIP1,YIP1) A statement function used to compute the ordinate value on the line joining the point (XI,YI) and the point (XIP1, YIP1) corresponding to the abscissa XX.

DXIN The incremental ordinate translation value to simulate stepping in the depth dimension.

DYIN The abscissa translation value similarly for simulated depth stepping.

FNSM1 The number of curves to be translated.

ABS(a) An intrinsic system function to calculate the absolute value of real number a.

INDEXT An index used for initializing the visual maximum function. Later used as an index during X-table lookups.

YZ A temporary variable used to compute the initial visual maximum function.

AMAX(a,b) An intrinsic system function used to determine the largest of two real numbers a and b.

EPS The incremental value added and subtracted from curve endpoint ordinate values.

DXKK	The x translation value to simulate depth stepping. Each curve is translated relative to the first curve displayed.
DYKK	Similarly, the y translation value for depth stepping.
RELINC	The slope of the x,y translation vector.
INDEXG	An index used for lookup of the XG table.
LAST	A flag for signaling that the end of X-table has been reached during lookup.
FIGURE	The array to hold scaled, packed data points for display.
IWHICH	A flag used in interval evaluation for hidden-line determinations.
SLOPE	The slope of the vector joining endpoints of the interval in the region where curve crosses visual maximum function.
NGRAPH	The number of points defining each visible curve segment.
NWD	The number of data words defining all visible portions of a single curve.

B. OPERATING INSTRUCTIONS

1. Graphics Terminal

The following should be entered on the terminal teletypewriter to execute POG3:

```
START("POG3",pvv)!
GATED!
```

where pvv is the pack and volume number location of POG3.

2. <u>SDS-9300 Computer</u>

The SENSE 2 button on the console should be depressed prior to reading in the card deck. This assures proper interpretation of the patch deck at the beginning of the program.

Immediately after the program starts execution, the user will notice an input light on the teletypewriter adjacent to the computer console. The proper operator response is:

IDEV=1* (carriage return)

This specifies AGT-10 number 1 for display. At present the variable control dials are available for use with unit number 1 only.

3. <u>Variable Control Dials</u>

a. Initial Settings

The following are suggested initial dial settings for use with the test function as listed in Appendix C:

DIAL	VALUE	RESULT
A and D	0.33	Determines an initial orientation.
B and E	-0.33	Positions the displayed structure approximately about the center of the CRT.
C	0.00 to 0.10	Allows continuance of succeeding curve display.
F	-0.99	Sets window to maximum negative value to suppress it's effects on the initial test function display.

b. View Orientation

To simulate rotation about the y axis, adjust DIAL A, positive values.

To simulate rotation about the x axis, adjust DIAL D, positive values.

c. Display Positioning

To modify horizontal (x) position, adjust DIAL B. To modify vertical (y) position, adjust DIAL E.

d. Window

Increasing the value of the window can be accomplished by increasing the setting of DIAL F.

e. Cursor

To display the cursor and accompanying text for a particular curve, set DIAL C greater than 0.10 immediately prior to display of the curve in sequence.

To modify x coordinates of cursor, adjust DIAL A.

To modify y coordinate of cursor variable endpoint, adjust DIAL D.

To exit from cursor mode and resume display of succeeding curves, set DIAL C less than or equal to 0.10.

APPENDIX C

COMPUTER PROGRAM

```
$PATCH
$>>DATA
007067  00106711
$END
      DIMENSION X(100),Y(100),Y1(100),XG(400),G(400),
     1     XH(400),H(400),IGDIR(51),DIAL(6)
      INTEGER FIGURE(101)
      DATA MAXDIM,XMIN,YMIN,XLNTH,YLNTH,XX/
     1     400,0.,-1.,-6.,-3.,3.14159265359/
      NAMELIST IDEV
      INPUT(101)
 9999 STEP = 3.14159265359/50.0
      CALL VCD(IDEV,IER,DIAL)
      IF (IER.NE.0) OUTPUT (101) IER,'   VCD'
      DELTAX = 5.*DIAL(1)
      IF (DIAL(1).LE.0.0001) DELTAX = 0.0001
      DELTAY = 5.*DIAL(4)
      IF (DIAL(4).LE.0.) DELTAY = 0.
      TX = 2.*DIAL(2)
      TY = 2.*DIAL(5)
      YWINDO = 2.*DIAL(6)
      IF (DIAL(3).LT.0.) GO TO 9999
      CALL DGINIT(IDEV,IGDIR,51,IER)
      IF(IER.NE.0) OUTPUT (101) IER,'  DGINIT'
      NG = -1
      N1 = 100
      NFNS = 30
      NUMFIG = 1
      X(1) = 0.
      Y1(1) = 0.
      DO 2 I = 2,N1
      X(I) = X(I-1) + STEP
    2 Y1(I) = .2*SIN(X(I))
      Z = 0.
      STEP = 3.14159265359/15.
```

```
      DO 3 I = 1,NFNS
      CZ = COS(Z)
      DO 4 J = 1,N1
    4 Y(J) = Y1(J)*CZ - (EXP(-(X(J) - XX)**2 - (Z - XX)**2)
     1*COS(1.75*((X(J) - XX)**2 + (Z - XX)**2)))*1.5
      CALL HIDE(X,Y,XG,G,XH,H,NG,MAXDIM,N1,NFNS,XLNTH,
     1      YLNTH,XMIN,DELTAX,YMIN,DELTAY,NUMFIG,IDEV,
     2      YWINDO,TX,TY)
    3 Z = Z + STEP
      GO TO 9999
      END
```

```
      SUBROUTINE HIDE(X,Y,XG,G,XH,H,NG,MAXDIM,N1,NFNS,
     1     XLNTH,YLNTH,XMIN,DELTAX,YMIN,DELTAY,NUMFIG,
     2     IDEV,YWINDO,TX,TY)
      DIMENSION X(1),Y(1),XG(1),G(1),H(1),XH(1),HDIAL(6)
      INTEGER FIGURE(101)
      DATA EPS1/.0000000001/
      F(XX,XI,YI,XIP1,YIP1) = YI + (XX - XI)*(YIP1 - YI)/
     1                        (XIP1 - XI)
      IF (MAXDIM.LE.0) RETURN
      DO 71 I = 2,N1
      IF (X(I - 1).LT.X(I)) GO TO 71
      MAXDIM = 0
      GO TO 75
   71 CONTINUE
      IF (NG.GT.0) GO TO 5000
      IF (N1 + 4.LE.MAXDIM) GO TO 74
      MAXDIM = -MAXDIM
   75 RETURN
   74 IF (NFNS.LE.0) GO TO 43
      FNSM1 = NFNS - 1
      DXIN = (9. - ABS(XLNTH))*DELTAX/FNSM1
      DYIN = (6. - ABS(YLNTH))*DELTAY/FNSM1
   43 INDEXT = 3
      DO 3 J = 1,N1
      XG(INDEXT) = X(J)
      YZ = Y(J) - EPS1
      G(INDEXT) = AMAX(YWINDO,YZ)
    3 INDEXT = INDEXT + 1
      EPS = EPS1*(ABS(XMIN) + ABS(DELTAX))
      NG = N1 + 4
      XG(1) = -FNSM1*DXIN + XMIN - ABS(XMIN) - ABS(XG(3))
     1        - 1.
      XG(2) = XG(3) - EPS
      XG(N1 + 3) = XG(N1 + 2) + EPS
      ZZ = YMIN
```

```
      G(1) = ZZ
      G(2) = ZZ
      G(N1 + 3) = ZZ
      G(NG) = ZZ
      DXKK = 0.
      DYKK = 0.
      RELINC = DELTAX/DELTAY
 5000 XG(NG) = X(N1)
      IF (NFNS) 52,48,49
   49 IF (NUMFIG.EQ.1) GØ TØ 48
      DXKK = DXKK + DXIN
      DYKK = DYKK + DYIN
   48 DØ 4 J = 1,N1
      Y(J) = Y(J) + DYKK
    4 X(J) = X(J) - DXKK
   52 CALL LØØKUP(X(1),XG(1),JJ)
      IF (JJ.GE.MAXDIM) GØ TØ 700
      DØ 31 J = 1,JJ
      XH(J) = XG(J)
   31 H(J) = G(J)
      IG = JJ + 1
      XH(IG) = X(1)
      H(IG) = F(X(1),XG(JJ),G(JJ),XG(IG),G(IG))
      INDEXG = JJ
      INDEXT = 1
      Z1 = X(1)
      F1 = H(IG) - Y(1)
      IT = 2
      JJ = IG
      IF (H(IG).GE.Y(1)) GØ TØ 32
      IF (JJ.GE.MAXDIM) GØ TØ 700
      JJ = IG + 1
      H(JJ) = Y(1)
      XH(JJ) = Z1 + EPS
   32 LAST = 0
```

```
      X1 = Z1
      FIGURE(1) = IHEAD(0,10)
      IZ = 2
 1100 IF (XG(IG).LT.X(IT)) GØ TØ 1001
      IWHICH = 0
      X2 = X(IT)
      F2 = F(X2,XG(IG - 1),G(IG - 1),XG(IG),G(IG)) - Y(IT)
      IT = IT + 1
      GØ TØ 1002
 1001 X2 = XG(IG)
      IWHICH = 1
      F2 = G(IG) - F(X2,X(IT - 1),Y(IT - 1),X(IT),Y(IT))
      IG = IG + 1
 1002 IF (F1*F?.GT.0.) GØ TØ 1005
      SLØPE = (F2 - F1)/(X2 - X1)
      IGG = IG - 1 - IWHICH
      ITT = IT - 2 + IWHICH
      IF (ABS(SLØPE*RELINC).GT.1.E-6) GØ TØ 1007
      Z2 = X2
      GØ TØ 1006
 1007 Z2 = X1 - F1/SLØPE
      GØ TØ 1006
 1005 X1 = X2
      F1 = F2
      IF (IT.LE.N1) GØ TØ 1100
 1008 LAST = 1
      Z2 = X(N1)
      CALL LØØKUP(Z2,XG(INDEXG),IGG)
      IGG = INDEXG + IGG - 1
      ITT = N1 - 1
 1006 ZZ = .99*Z1 + .01*Z2
      CALL LØØKUP(ZZ,X(INDEXT),K1)
      CALL LØØKUP(7Z,XG(INDEXG),<2)
      K1 = K1 + INDEXT - 1
      K2 = K2 + INDEXG - 1
```

```
      IF (F(ZZ,X(K1),Y(K1),X(K1 + 1),Y(K1 + 1)).GT.
     1    F(ZZ,XG(K2),G(K2),XG(K2 + 1),G(K2 + 1))) GØ TØ 7
      IF (JJ + IGG - INDEXG.GE.MAXDIM) GØ TØ 700
      IF (INDEXG.EQ.IGG) GØ TØ 712
      J1 = INDEXG + 1
      DØ 12 I = J1,IGG
      JJ = JJ + 1
      XH(JJ) = XG(I)
   12 H(JJ) = G(I)
  712 JJ = JJ + 1
      XH(JJ) = Z2
      H(JJ) = F(Z2,XG(IGG),G(IGG),XG(IGG + 1),G(IGG + 1))
      INDEXG = IGG
      INDEXT = ITT
      GØ TØ 60
    7 NGRAPH = ITT - INDEXT + 2
      IF (JJ + NGRAPH - 1.GT.MAXDIM) GØ TØ 700
      N2 = JJ
      IF (NGRAPH.EQ.2) GØ TØ 9
      J1 = INDEXT + 1
      DØ 11 I = J1,ITT
      JJ = JJ + 1
      XH(JJ) = X(I)
   11 H(JJ) = Y(I)
    9 JJ = JJ + 1
      XH(JJ) = Z2
      H(JJ) = F(Z2,X(ITT),Y(ITT),X(ITT + 1),Y(ITT + 1))
      XF = XH(N2)/5. + TX
      YF = H(N2)/5. + TY
      FIGURE(IZ) = IPACK(XF,YF,0)
      MM = N2 + NGRAPH - 1
      N3 = N2 + 1
      IZ = IZ + 1
      DØ 9001 M = N3,MM
      XF = XH(M)/5. + TX
```

```
      YF = H(M)/5. + TY
      FIGURE(IZ) = IPACK(XF,YF,1)
 9001 IZ = IZ + 1
 9000 INDEXT = ITT
      INDEXG = IGG
   60 IF (LAST.EQ.1) GØ TØ 61
      X1 = X2
      F1 = F2
      Z1 = Z2
      IF (IT.LE.N1) GØ TØ 1100
      GØ TØ 1008
   61 NWD = IZ - 1
      IF (NWD.LE.1) GØ TØ 62
      DØ 63 J = IZ,101
   63 FIGURE(J) = 0
      CALL GRAPHØ(IDEV,FIGURE(1),101,NUMFIG,IER)
      IF (IER.NE.0) ØUTPUT (101) IER,NUMFIG
      NUMFIG = NUMFIG + 1
   62 IF (XG(NG).LE.XG(NG - 1)) NG = NG - 1
      IF (XG(NG).LE.X(N1)) GØ TØ 33
      IF (JJ + 3 + NG - IGG.GT.MAXDIM) GØ TØ 700
      XH(JJ + 1) = XH(JJ) + EPS
      JJ = JJ + 1
      H(JJ) = F(X(N1),XG(IGG),G(IGG),XG(IGG + 1),G(IGG + 1))
      IGGP1 = IGG + 1
      DØ 34 J = IGGP1,NG
      JJ = JJ + 1
      XH(JJ) = XG(J)
   34 H(JJ) = G(J)
   33 NG = JJ + 2
      IF (NG.GT.MAXDIM) GØ TØ 700
      YWINDØ = YWINDØ + DYIN
      DØ 13 I = 1,JJ
      G(I) = AMAX(YWINDØ,H(I))
   13 XG(I) = XH(I)
```

```
      XG(JJ + 1) = XG(JJ) + EPS
      G(JJ + 1) = YMIN + DYKK
      G(NG) = G(JJ + 1)
   66 IF (NFNS.LT.0) GØ TØ 53
      DØ 82 I = 1,N1
      X(I) = X(I) + DXKK
   82 Y(I) = SIGN*Y(I) - DYKK
   53 CALL VCD(IDEV,IER,HDIAL)
      IF (IER.NE.0) ØUTPUT (101) IER,'   VCD2'
      IF (HDIAL(3).GT.0.1) CALL CURSØR(DXKK,DYKK,NUMFIG,
     1                             IDEV,TX,TY)
      RETURN
  700 MAXDIM = -MAXDIM
      GØ TØ 66
      END
```

```
      SUBROUTINE LOOKUP(X,XTBL,J)
      DIMENSION XTBL(1)
      J = 2
    4 IF (XTBL(J) - X) 1,2,3
    1 J = J + 1
      GO TO 4
    2 RETURN
    3 J = J - 1
      RETURN
      END
```

```
      SUBROUTINE CURSOR(DXKK,DYKK,NUMFIG,IDEV,TX,TY)
      DIMENSION CDIAL(6),ICURS(101)
      DIMENSION ITDIR(7),ITXT1(4),ITXT2(2)
      ICURS(1) = IHEAD(0,10)
      DO 5 I = 4,101
    5 ICURS(I) = 0
      LNUM = NUMFIG - 1
      CALL DTINIT(IDEV,ITDIR,7,IER)
      ENCODE(16,20,ITXT1)
   20 FORMAT('LINE NUMBER =    ')
      CALL TEXT0(IDEV,ITXT1,4,36,1,1,3,IER)
      ENCODE(4,30,ITXT3) LNUM
   30 FORMAT(I4)
      CALL TEXT0(IDEV,ITXT3,1,36,13,1,3,IER)
      ENCODE(4,40,ITXT1)
   40 FORMAT('X = ')
      CALL TEXT0(IDEV,ITXT1,1,38,1,1,3,IER)
      ENCODE(4,50,ITXT1)
   50 FORMAT('Y = ')
      CALL TEXT0(IDEV,ITXT1,1,40,1,1,3,IER)
   99 CALL VCD(IDEV,IER,CDIAL)
      IF (IER.NE.0) OUTPUT (101) IER,'VCDCUR'
      CX1 = 2.*CDIAL(1) - DXKK/5. + TX
      ICURS(2) = IPACK(CX1,DYKK/5. + TY, 0)
      ICURS(3) = IPACK(CX1,2.*CDIAL(4) + DYKK/5. + TY,1)
      CALL GRAPH0(IDEV,ICURS(1),101,NUMFIG,IER)
      IF (IER.NE.0) OUTPUT (101) IER,'CURSOR',NUMFIG
      CALL UNPACK(ICURS(3),CX2,CY2,IMD)
      X = (CX2 - TX)*5. + DXKK
      Y = (CY2 - TY)*5. - DYKK
      ENCODE(8,60,ITXT2) X
   60 FORMAT(F8.2)
      CALL TEXT0(IDEV,ITXT2,2,38,5,1,3,IER)
      ENCODE(8,70,ITXT2) Y
   70 FORMAT(F8.2)
```

```
CALL TEXTØ(IDEV,ITXT2,2,40,5,1,3,IER)
IF (CDIAL(3).GT.0.1) GØ TØ 99
CALL DTINIT(IDEV,ITDIR,7,IER)
RETURN
END
```

```
$VCD    PZE    0
        BRM    9SETUPN
        PZE    3
        PZE    0
        PZE    0
        PZE    0
        LDA    *VCD+3
        BRM    DVN8CK
        BRU    VCDDER
        STA    VCDCL+2
        STA    FI8BFL
        LDA    =035000000
        STA    VCDSW8
        EOM    032004
VCDCL   BRM    D\EXEC
        PZE    2
        PZE    0
        PZE    VCDSW8
        EOM    032001
        EOM    032002
        LDA    *VCD+3
        SKA    I8BFL
        BRU    $-1
        LDX    =0177772,1
VCDAG   LDA    077756,1
        LDB    =0
        FLA    Q0
        STD    *VCD+5
        MPT    VCD+5
        BRX    VCDAG,1
        LDA    =0
        BRU    VCDER
VCDDER   LDA   =1
VCDER   BRR    VCD
VCDSW8   PZE   0
```

```
        PZE     0
        PZE     0
QQ      PZE     0
        END
```

APPENDIX D

CONTROL DIAL IMPLEMENTATION

An initial change was made to GATED [Ref. 17], the AGT-10 program to display and edit graphics and text data blocks. The alteration included a call to AGT-10 system subroutine TRVCD [Ref. 19], which obtains the digitized values of the output voltages of the variable control dials. The sampling and storage of the six-dial values requires 750 micro-seconds. The sub-program, which includes the call to TRVCD, was titled POGG. It thus became a required subprogram of POG3, the revised version of GATED.

In addition to sampling the control dial settings, POGG transmitted the six values to the SDS-9300 computer and stored them in upper-core locations. POGG was invoked once every cycle through POG3. This meant that current dial values were available every 0.025 seconds.

Once the dial values were resident in the 9300 computer, they were transferred to locations designated by a FORTRAN array name, e.g., DIAL. At this point scaling of the dial setting was completed and the resultant values passed to subroutine HIDE for display purposes.

An intermittent difficulty occurred with control dial testing. The problem manifested itself as sudden and apparently random blanking of CRT display. Eventual diagnosis was interference between sending control dial information, from the AGT to the 9300 computer, and coincident transmission from the computer to the graphics terminal of a graphics display request, GRAPHO [Ref. 18].

Solution of the interference problem involved the use of a meta-symbol subroutine, VCD. The purpose of VCD was to limit the sending of

APPENDIX D

CONTROL DIAL IMPLEMENTATION

An initial change was made to GATED [Ref. 17], the AGT-10 program to display and edit graphics and text data blocks. The alteration included a call to AGT-10 system subroutine TRVCD [Ref. 19], which obtains the digitized values of the output voltages of the variable control dials. The sampling and storage of the six-dial values requires 750 micro-seconds. The sub-program, which includes the call to TRVCD, was titled POGG. It thus became a required subprogram of POG3, the revised version of GATED.

In addition to sampling the control dial settings, POGG transmitted the six values to the SDS-9300 computer and stored them in upper-core locations. POGG was invoked once every cycle through POG3. This meant that current dial values were available every 0.025 seconds.

Once the dial values were resident in the 9300 computer, they were transferred to locations designated by a FORTRAN array name, e.g., DIAL. At this point scaling of the dial setting was completed and the resultant values passed to subroutine HIDE for display purposes.

An intermittent difficulty occurred with control dial testing. The problem manifested itself as sudden and apparently random blanking of CRT display. Eventual diagnosis was interference between sending control dial information, from the AGT to the 9300 computer, and coincident transmission from the computer to the graphics terminal of a graphics display request, GRAPHO [Ref. 18].

Solution of the interference problem involved the use of a meta-symbol subroutine, VCD. The purpose of VCD was to limit the sending of

control dial values to the SDS-9300 computer. Values were then sent only after VCD had been called. This interrupt condition insured that no conflicts would occur.

The inclusion of the metasymbol subroutine VCD forced further modificiations of POG3. A patch deck was necessary at the beginning of the program. The patch deck used is included in the program listing, Appendix C. A test for code 35, the designation given to a request for control dial sensing in POG3, was included. Additionally, changes to instructions in AXINT, a required sub-program of POG3, were made to preclude a jump to the magnetic tape operation. This operation had the only other 30 series request code. A SETGT instruction was substituted for several consecutive operators in AXINT to allow for code 35.

POG3 is currently available on the user file pack for AGT-1 in the Naval Postgraduate School Computer Laboratory. Instructions for loading and executing POG3 are included in Appendix B. Following is a summary of programming changes to GATED.

1. Title changed to POG3.
2. After third instruction under label SA3 and immediatly after:

```
JPAN        SB
```

the following code was added:

```
MDAR'H      $GTWD1
ARRS        3
MDAR'F'A    77
MDXO'F      35
JPLS        .+5
JPSR        $POGG
JPSR        FSWI
            VBLK
JUMP        SB
```

The code sequence resumes as in GATED with

```
MDAR'H      $GTWD1
```

3. After label 6BLK: 200!H, add

```
VBLK:       350!H
```

4. File program POGG required by POG3 is as follows:

```
EXPUNGE
TITLE       POGG
ENTRY       POGG
POGG:       JUMP          .
            JPSR          $TRVCD
            ARXO'F
            ARMD          CNT
            MDAR'F        $TVCDA
            MDAE'L;       -1
FADD:       0
            MDAR'X'I      FADD
            ARRS          6
            NOOP
            ARMD'I        FADD
            MDAR'X        CNT
            MDXO'L;       6
            JPLS          FADD+1
            JPSR          $ROWFW
                          -0
                          POGG1
                          $TVCDA
                          6
            MDIR          POGG
POGG1:      77750
CNT:        0
TERMINATE
```

BIBLIOGRAPHY

1. Appel, A., "The Notion of Quantitative Invisibility and the Machine Rendering of Solids," Proceedings of the 22nd ACM National Conference, pp. 387-393, 1967.

2. Comba, P.G., "A Procedure for Detecting Intersections of Three-Dimensional Objects," Journal of the ACM, v. 15, p. 354-366, July 1968.

3. Desens, R.B., Computer Processing for Display of Three-Dimensional Structures, E.E. Thesis, Naval Postgraduate School, 1969.

4. Freeman, H. and Loutrel, P.P., "An Algorithm for the Solution of the Two-Dimensional Hidden-Line Problem," IEEE Transactions on Electronic Computers, v. EC-16, pp. 784-790, December 1967.

5. Galimberti, R. and Montanari, B., "An Algorithm for Hidden-Line Elimination," Communications of the ACM, v. 12, pp. 206-211, April 1969.

6. Hagan, T.C. and Treiber, R., "Hybrid Analog/Digital Techniques for Signal Processing Applications," Proceedings of the Spring Joint Computer Conference, v. 28, pp. 379-388, 1966.

7. Hagan, T.G., Nixon, R.J., and Schaefer,L.J., "The Adage Graphics Terminal," Proceedings of the Fall Joint Computer Conference, v. 33, pp. 747-755, 1968.

8. Johnson, T.E., "Sketchpad III-A Computer Program for Drawing in Three Dimensions," Proceedings of the Spring Joint Computer Conference, v.22, pp. 347-353, 1963.

9. Kubert, B., Szabo, J., and Giulieri, S., "The Perspective Representation of Functions of Two Variables," Journal of the ACM, v. 15, p. 193-204, April 1968.

10. Matsushita, Y., "Hidden Lines Elimination for a Rotating Object," Communications of the ACM, v. 15, pp. 245-252, April 1972.

11. Naval Postgraduate School Computer Laboratory, Library Program Report, DTINIT, by W.A. Thomas, p. 1, 2 September 1970.

12. Naval Postgraduate School Computer Laboratory, Library Program Report, IHEAD, by R. L. Limes, p. 1, 31 August 1970.

13. Naval Postgraduate School Computer Laboratory, Library Program Report, IPACK, by R. L. Limes, p. 1, 9 September 1970.

14. Naval Postgraduate School Computer Laboratory, Library Program Report, UNPACK, by R.L. Lines, p. 1, 9 September 1970.

15. Naval Postgraduate School Computer Laboratory, Library Program Report, TEXTO, by W. A. Thomas, p. 1, 2 September 1970.

16. Naval Postgraduate School Computer Laboratory, Program Report, DGINIT, by W.A. Thomas, p. 1-4, 8 May 1972.

17. Naval Postgarduate School Computer Laboratory, Program Report, GATED, by W. A. Thomas, p. 1-4, 8 May 1972.

18. Naval Postgraduate School Computer Laboratory, Program Report, GRAPHO, by W. A. Thomas, p. 1-5, 5 April 1972.

19. Naval Postgraduate School Computer Laboratory, Program Report, TRVCD, by W. A. Thomas, p. 1-2, 16 February 1972.

20. Puckett, H.R., "Computer Method for Perspective Drawing," Journal of Spacecraft and Rockets, v. 1, pp. 44-48, January 1964.

21. Sutherland, I.E., "Sketchpad, A Man-Machine Graphical Communication System," Proceedings of the Spring Joint Computer Conference, v. 22, pp. 329-346, 1963.

22. Williamson, H., "Hidden-Line Plotting Program," Communications of the ACM, v. 15, pp. 100-103, February 1972.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Assoc. Professor G. A. Rahe, Code 52 Ra Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	LT Steven Neil Poggi, USN 641 Kinley Street La Habra, California 90631	1

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE
Hidden-Line Elimination with Interactive Graphics Display Considerations

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)
Master's Thesis (June 1972)

5. AUTHOR(S) (First name, middle initial, last name)
Steven Neil Poggi

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1972	69	22

8a. CONTRACT OR GRANT NO. b. PROJECT NO. c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT
Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

Problem areas in the field of computer graphics, as applied to three-dimensional space, are introduced through a discussion of hidden-line elimination and perspective views. An adaptation of a simple, fast algorithm for hidden-line elimination is presented.

Graphics terminal display capability for a two-dimensional representation of a surface is made available. Interactive extensions to the basic program are developed to enhance potential applications for design and analysis.

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Hidden-Line Elimination						
Interactive Graphics						

DD FORM 1 NOV 65 1473 (BACK)

S/N 0101-807-6821

Security Classification

A-31409

Thesis 136280
P665 Poggi
c.1 Hidden-line elimination with interactive graphics display consideration.